Ralf Hauser & Angelika Storrer

Probleme und Lösungen beim Parsen von Wörterbüchern

1 Einleitung

Viele neuere Wörterbücher liegen maschinenlesbar nur als Satzbanddateien vor. Diese Satzbanddateien werden bei der Drucklegung des Wörterbuchs vom Setzer mittels einer Satzmaschine erstellt. Ein Wörterbuchparser ist ein Programmsystem, das einen auf Satzband repräsentierten Wörterbuchtext maschinell in seine zentralen Gliederungseinheiten, die Wörterbuchartikel, zerlegt und diesen Wörterbuchartikeln dann relativ zu einer vorgegebenen Grammatik eine Strukturbeschreibung zuweist. Der Wörterbuchparser erzeugt dabei eine Repräsentation des Wörterverzeichnis, die die hierarchische Struktur der einzelnen Wörterbuchartikel expliziert und damit einen gezielten Zugriff auf die einzelnen lexikographischen Angaben ermöglicht. Eine derartige Repräsentation der Wörterbuchstruktur wird im folgenden als *Wörterbuchdatenbank* bezeichnet und kann in verschiedener Weise genutzt werden:

- In der **praktischen Lexikographie** können Wörterbuchdatenbanken als zentraler Bestandteil eines computerunterstützten Lexikographenarbeitsplatzes fungieren: Die als Datenbank gespeicherten Wörterbücher sind dann direkt am Bildschirm editier- und veränderbar. Die vielfältigen Abfragemöglichkeiten gestatten einen raschen und effizienten Zugriff auf sämtliche Wörterbuchartikel und erleichtern dadurch die Überprüfung der strukturellen und inhaltlichen Konsistenz des Wörterbuchs. Außerdem können sie als Grundlage für künftige elektronische Publikationen von Wörterbüchern verwendet werden (vgl. Bläser/Wermke 1990).
- In der **Computerlinguistik** gab es in den letzten Jahren verschiedene Versuche, das in Wörterbuchdatenbanken gespeicherte lexikalische Wissen für Systeme der maschinellen Sprachverarbeitung nutzbar zu machen. Dabei wurden entweder Sprachverarbeitungsprogramm und Wörterbuchdatenbank durch eine Schnittstelle direkt miteinander verbunden, oder es wurden spezifische Informationen aus der Wörterbuchdatenbank (semi-)automatisch extrahiert und in eine Datenstruktur überführt, die dann von einem oder mehreren Sprachverarbeitungsprogrammen genutzt werden kann (z.B. Boguraev 1991, McNaught 1988 und Boguraev/Briscoe 1989). Die Versuche sind vornehmlich durch das Bedürfnis motiviert, den hohen Zeit- und Kostenaufwand zu senken, der mit der Entwicklung von Lexikonkomponenten für Sprachverarbeitungssysteme verbunden ist.

Die Extrahierbarkeit lexikalischen Wissens aus Wörterbuchdatenbanken ist auch Teil des Forschungsinteresses in dem an der Universität Tübingen durchgeführten ELWIS-Projekt zum korpusgestützten Aufbau lexikalischer Wissensbasen (vgl. Storrer/Feldweg/Hinrichs 1993). Für Experimente zur Extraktion lexikalischen Wissens standen dem Projekt verschiedene lexikalische Datensammlungen zur Verfügung, u.a. Satzbanddateien mit Teilen

des Duden-Stilwörterbuchs und des Duden-Bedeutungswörterbuchs.¹ Um auf die für das Projekt relevanten lexikographischen Angaben zu Flexion, Wortbildung und zum kombinatorischen Potential lexikalischer Einheiten (Subkategorisierung, Kollokationen, Phraseolexeme) gezielt zugreifen zu können, mußten diese Satzbanddateien in eine Wörterbuchdatenbank überführt werden. Zu diesem Zweck wurde das LEXPARSE-System entwickelt, ein flexibler und wörterbuchunabhängiger Wörterbuchparser, mit dem beliebige Satzbanddateien geparkt werden können.

Im Gegensatz zu vergleichbaren Systemen² basiert LEXPARSE nicht auf einem Parser für natürlichsprachige Sätze, sondern wurde von Beginn an speziell auf die Erkennung von Strukturen konzipiert, wie sie für standardisierte Wörterbuchtexte charakteristisch sind. LEXPARSE verfügt über eine benutzerkonfigurierbare Ausgabekomponente, in der die Artikelstruktur in verschiedener Weise dargestellt werden kann. Das System fördert durch Protokollmechanismen und Testhilfen eine zügige und komfortable Grammatikentwicklung und verarbeitet einen kontextfreien Grammatikformalismus mit problemspezifischen Erweiterungen. Den metalexikographischen Hintergrund bei der Systementwicklung lieferte die metalexikographische Texttheorie von H.E. Wiegand (vgl. Wiegand 1989b, Wiegand 1989c, Wiegand 1991), die sowohl eine detaillierte Klassifikation von lexikographischen Textsegmenten als auch die theoretische und methodische Basis zur Entwicklung von Artikelstrukturgrammatiken bereitstellt.

Im zweiten Abschnitt werden wir Aufgabenstellung und Problematik des Wörterbuchparsings an einem Beispiel aus dem Duden-Stilwörterbuch erläutern. Wichtig ist hierbei der Unterschied zwischen der intellektuellen und der maschinellen Strukturanalyse der Wörterbuchartikel: Eine menschliche Benutzerin kann hierbei auf ihr Sprach- und Weltwissen zurückgreifen und ihr Vorwissen über Wörterbuchstrukturen einsetzen. Ein solches Wissen steht einem Wörterbuchparser bei der Analyse der Satzbandrepräsentation des Wörterbuchs nicht zur Verfügung. Dieser ist vielmehr bei der Strukturerkennung auf das Vorhandensein und die eindeutige Interpretierbarkeit von explizit strukturanzeigenden Symbolen (Steuerzeichen, die eine Schriftartenumschaltung anzeigen; Sonderzeichen mit strukturanzeigender Funktion) angewiesen. Hieraus ergeben sich die drei Kernprobleme des Wörterbuchparsings:

- Die Identifikation der relevanten lexikographischen Textsegmente.
- Die Auflösung funktionaler Ambiguitäten bei strukturanzeigenden Symbolen.
- Die Verarbeitung von typographischer Information.

Im dritten Abschnitt werden wir das Systemdesign von LEXPARSE skizzieren und dann zeigen, wie die einzelnen Komponenten bei der Lösung dieser drei Kernprobleme des Wörterbuchparsings zusammenspielen. Dabei steht die informelle Darstellung der Probleme und der in LEXPARSE realisierten Lösungsstrategien im Vordergrund dieses Artikels; für eine ausführliche Darstellung der Gesamtfunktionalität des Systems, dessen Implementierung und eine Diskussion des vom System unterstützten Grammatikformalismus verweisen wir auf Hauser 1993, Hauser/Storror 1993 und Hauser/Storror 1994.

1 Wir möchten uns bei Dr. Matthias Wermke und der Duden-Redaktion dafür bedanken, daß sie uns diese Datenbestände zur Verfügung gestellt haben.

2 Hier ist vor allem der *Dictionary Entry Parser* zu nennen, ein in Prolog implementiertes System, das im Forschungszentrum der IBM in Yorktown Heights entwickelt wurde (vgl. Neff/Boguraev 1990).

2 Wörterbuchparsing: Aufgabenstellung und Probleme

In der modernen metalexikographischen Forschung werden standardisierte Wörterbücher als Texte beschrieben, deren Konstituenten in komplexer Weise miteinander verbunden sind. Die für das Wörterverzeichnis eines Wörterbuchs spezifischen Strukturen können mit dem theoretischen und methodischen Instrumentarium der lexikographischen Texttheorie von H.E. Wiegand analysiert und repräsentiert werden (vgl. Wiegand 1989b und Wiegand 1991). Die dort vorgeschlagenen Varianten der funktional–positionalen Segmentierungsmethode liefern die Konstituenten der Wörterbuchartikel; diese können dann hinsichtlich ihrer Beziehungen zueinander untersucht werden. Für die Belange des Wörterbuchparsings sind vor allem Teil–Ganzes– und Abfolgebeziehungen zwischen diesen Konstituenten relevant, die gemeinsam die hierarchische Mikrostruktur eines Wörterbuchartikels determinieren.

Die Abbildung im Anhang auf Seite 65 zeigt die konkrete hierarchische Mikrostruktur des Lemmas *Treffen* aus dem Duden-Stilwörterbuch, wie sie von LEXPARSE ausgegeben wird, wenn im Ausgabeformat eine Darstellung als Attribut–Werte–Baum gewählt wird.³ Der Wurzelknoten **WA** (*Wörterbuchartikel*), der in der zugehörigen LEXPARSE–Grammatik als Startsymbol fungiert, hat als direkte Konstituenten die drei lexikographischen Kommentare **FK** (*Formkommentar*), **SK** (*Semantischer Kommentar*) und **PKP** (*Postkommentar zur Phraseologie*). Der semantische Kommentar (**SK**) verzweigt weiter in verschiedene semantische Subkommentare, in denen die einzelnen polysemen Lesarten des Lemmazeichens abgehandelt werden.

Das Duden-Stilwörterbuch unterscheidet drei Arten von *semantischen Subkommentaren* (**SSK0**, **SSK1**, **SSK2**) durch verschiedene Arten von Zählern: Römische Zahlen indizieren Homographen, arabische Zahlen trennen polyseme Lesarten und Kleinbuchstaben trennen unterschiedliche Bedeutungsvarianten voneinander ab. Die lexikographischen Kommentare setzen sich aus einer Folge von lexikographischen Angaben bestimmter Klassen zusammen; welche Klassen von Angaben potentiell zu einem Lemmazeichen gemacht werden können, ist in Abhängigkeit von der Wortart des Lemmazeichens im Mikrostrukturprogramm eines Wörterbuchs festgelegt. Dabei unterscheiden wir Klassen komplexer Angaben, die noch weiter zerlegt werden können (z.B. besteht die *Beispielgruppenabgabe* (**BeiGA**) aus einer Folge von *Beispielangaben* (**BeiA**)), von Klassen elementarer, nicht weiter zerlegbarer Angaben (z.B. die Klasse der *pragmatischen Angaben* (**PragA**) oder die Klasse der *Beispielangaben* (**BeiA**)). Diesen sind dann die eigentlichen Angabetexte zugeordnet, die in der vorliegenden Notation durch Anführungsstriche gekennzeichnet werden.

Angabetexte sind die terminalen Einheiten der hierarchischen Mikrostrukturen von Wörterbuchartikeln und können ihrerseits aus einer Folge von Wörtern und Zeichen bestehen, ja sogar eine interne syntaktische Struktur besitzen. Dennoch kann eine Wörterbuchbenutzerin im allgemeinen unter Einsatz ihres Sprach– und Weltwissens die relevanten lexikographischen Textsegmente identifizieren, zumal wenn sie über genügend Vorerfahrung im Umgang mit standardisierten Wörterbüchern verfügt. In Zweifelsfällen können die Benutzerhinweise im Vorspanntext des Wörterbuchs zu Rate gezogen werden. Auch die funktional–positionale Segmentierung, als Methode zur wissenschaftlichen Analyse von Wörterbuchstrukturen, greift auf dieses Wissen zurück.

3 Äquivalente Repräsentationen als SGML–Struktur oder in einer Pfad–Notation können ebenfalls erzeugt werden; vgl. die Beispiele in Hauser/Storrer 1993.

Treffen, das: 1. *Zusammenkunft, Begegnung*: regelmäßige, seltene T.; ein T. der Abiturienten; ein T. der Außenminister; ein T. verabreden, veranstalten; an einem T. teilnehmen; zu einem T. kommen. 2. (militär. veraltet) *Gefecht*: frische Truppen ins T. führen. 3. (Sport) *Wettkampf*: ein faires, spannendes T.; das T. endete unentschieden; sie konnte das T. für sich entscheiden. * (geh. :) *etwas ins Treffen führen* (*etwas als Argument vorbringen*).

Abbildung 1: Ausschnitt aus dem Duden-Stilwörterbuch (Lemma: *Treffen*; S. 703)

Ein Wörterbuchparser hingegen muß die Struktur von Wörterbuchartikeln ohne Rückgriff auf Sprach- und Weltwissen analysieren. Die korrekte Segmentierung des Wörterbuchtextes in die einzelnen lexikographischen Textsegmente stellt dabei eines der Kernprobleme dar: Herkömmliche Tokenisierungsstrategien führen aufgrund der Mehrgliedrigkeit der terminalen Einheiten nicht unmittelbar zu den elementaren lexikographischen Textsegmenten. Diese müssen vielmehr erst aus ihren Bestandteilen zusammengesetzt und dann einer bestimmten Angabeklasse zugeordnet werden. Hierbei stellt sich das zusätzliche Problem, daß die meisten Angabeklassen nicht über einen festen Wertebereich verfügen: Angabeklassen wie *BeiA* (*Beispielangaben*), *PhrasA* (*Phrasenangaben*) sind relativ frei bildbare Texte, für die keine endliche Menge zulässiger Elemente in einem 'Angabenlexikon' spezifiziert werden kann. Deshalb kann, im Unterschied zum herkömmlichen Vorgehen beim Parsen von natürlichsprachlichen Sätzen, die Zuordnung eines lexikographischen Textsegments zu einer der im Mikrostrukturprogramm vorgesehenen Angabeklassen auch nicht über einen Nachschlageprozess in einem solchen Lexikon erfolgen. Vielmehr muß beides — sowohl die Segmentierung des Wörterbuchtextes in elementare und komplexe lexikographische Textsegmente, als auch die Zuordnung der elementaren Textsegmente zu den entsprechenden Angabeklassen — durch die korrekte Analyse von strukturanzeigenden Symbolen gesteuert werden.

Abbildung 1 (S. 56) zeigt den Wörterbuchartikel für das Lemma *Treffen*, wie es im Stilwörterbuch abgedruckt ist. Auf der Ebene des gedruckten Wörterbuchtextes lassen sich zwei Arten von Strukturanzeigern unterscheiden:

Nicht-typographische Strukturanzeiger sind Symbole, die als Teile des Wörterbuchtextes die hierarchische Mikrostruktur verdeutlichen, indem sie Angaben voneinander separieren und als Elemente bestimmter Angabeklassen kenntlich machen. Im Stilwörterbuch wird beispielsweise, wie im Artikel zu *Treffen* sichtbar, der *Formkommentar* (FK) vom *semantischen Kommentar* (SK) durch den Strukturanzeiger Doppelpunkt ':' getrennt; der Strukturanzeiger Stern '*' separiert den *Semantischen Kommentar* vom *Postkommentar zur Phraseologie* (PKP). Zwischen verschiedenen *Beispielangaben* (*BeiA*) innerhalb einer *Beispielgruppenangabe* (*BeiGA*) fungiert der Strukturanzeiger Semikolon ';' als Separator.

T2ñi5ñTreffen, i1ñdas: i3ñ1. i2ñZusammenkunft, Begegnung:ivr10ñ }Ã}
 i1ñregelm"Σige, seltene T.; ein T. der Abiturienten; }Ã}
 ein T. der AuΣenminister; ein T. verabreden, ver}-}
 anstalten; an einem T. teilnehmen; zu einem T. }A}
 kommen. i3ñ2. i1ñ(milit"r. veraltet) i2ñGefecht:ivr10ñ i1ñfrische }Ã}
 Truppen ins T. f'hren. i3ñ3. i1ñ(Sport) i2ñWettkampf:ivr10ñ i1ñein }Ã}
 faires, spannendes T.; das T. endete unentschie)-}
 den; sie konnte das T. f'r sich entscheiden. }Ã}
 iRT20ñ*iRTñπkz(geh.:) i3ñetwas ins Treffen f'hren i2ñ(etwas als Argu)-}
 ment vorbringen).ivr10ñë

Abbildung 2: Ausschnitt aus der Satzbanddatei für das Duden-Stilwörterbuch (Lemma: *Treffen*)

Typographische Strukturanzeiger sind z.B. die Schriftarten und Schnitte, in denen die Angaben bestimmter Klassen erscheinen. So erscheint in unserem Beispiel die *Lemmazeichengestaltangabe* (LZGA) und die *Phrasemangabe* (PhrasA) in Fettdruck, die *Bedeutungsparaphrasenangabe* (BPA) in Kursivdruck.

Strukturanzeiger unterstützen die kundige Wörterbuchbenutzerin dabei, sich im Wörterbuchartikel zu orientieren und erleichtern einen raschen und gezielten Zugriff auf die gesuchte Information.

Auf der Ebene des gedruckten Wörterbuchtextes treten typographische Strukturanzeiger lediglich indirekt in Erscheinung, d.h. als charakteristische Eigenschaft einer Angabe oder eines Strukturanzeigers. In den Satzbanddateien hingegen, die die Eingabe für den Wörterbuchparser bilden, wird typographische Information direkt repräsentiert: Abbildung 2 (S. 57) zeigt die Satzbandrepräsentation für den Wörterbuchartikel zum Lemma *Treffen* im Duden-Stilwörterbuch. Eine Analyse dieser Satzbandrepräsentation ergibt, daß die Steuerzeichen δ1û und δ2û als Umschaltung in die Schriftschnitte *fett* bzw. *kursiv* zu interpretieren sind. Ebenso gibt es Steuerzeichen, die eindeutig Beginn und Ende eines Wörterbuchartikels identifizieren, und mit deren Hilfe der Satzbandtext relativ problemlos in die einzelnen Wörterbuchartikel segmentiert werden kann.

Die Hauptschwierigkeit des Wörterbuchparsers ist es dann, die hierarchische Struktur dieser Wörterbuchartikel zu analysieren und diese in einem geeigneten Format auszugeben. Hierbei spielt die korrekte Interpretation der Strukturanzeiger und der typographischen Steuerzeichen eine zentrale Rolle. Neben den strukturelevanten Steuerzeichen finden sich im Satzbandtext auch strukturirrelevante Sequenzen (die z.B. Zeilen- oder Seitenumbruch bewirken), sowie Kopfzeilen, Seitennummern und andere layoutspezifische Zeichenfolgen; diese müssen vom Wörterbuchparser gelöscht oder ignoriert werden können.

Im folgenden Abschnitt wird gezeigt, wie die verschiedenen Module des Systems LEX-PARSE, das speziell für die maschinelle Erkennung von Wörterbuchstrukturen konzipiert wurde, bei der Bewältigung dieser komplexen Aufgabenstellung zusammenspielen.

3 Wörterbuchparsing mit LEXPARSE

Der schematische Aufbau des LEXPARSE-Systems wird durch Abbildung 3 veranschaulicht.⁴

Abbildung 3: Der Aufbau des LEXPARSE Programms (Quelle: Hauser 1993)

Das LEXPARSE-System setzt sich aus verschiedenen Subsystemen zusammen:

- aus einem Präprozessor,
- einem Scanner,
- dem Parser,

4 Der funktionale Aufbau des Programms spiegelt sich auch in der Objekthierarchie wieder, die der objektorientierten Implementierung zugrundeliegt.

- und einer Ausgabekomponente, die den erzeugten Ableitungsbaum auf verschiedene Arten darstellen kann.

Der **Präprozessor** liest zeilenweise aus dem Satzband und nimmt eine Umsetzung der satzbandspezifischen Formatangaben und Steuerzeichen vor. Dabei werden nicht benötigte Steuerzeichen verworfen und andere durch Textelemente ersetzt (z.B. bei der Umsetzung der deutschen Umlaute).

Der **Scanner** zerlegt die umgewandelten Zeilen des Präprozessors in einzelne atomare Zeichenketten, die sog. Tokens. LEXPARSE unterscheidet zwei Klassen von Tokens, die sog. *Literal-Tokens* und die sog. *XCode-Tokens*.

Der **Parser** überprüft die vom Scanner gelieferten Tokens und entscheidet anhand der Grammatik, welche Regeln zu expandieren sind. Dabei wird ein Ableitungsbaum konstruiert, der die Struktur des Wörterbuchartikels expliziert.

Der **Postprozessor** kann daran anschließend noch einmal textuelle Umsetzungen am Ableitungsbaum vornehmen. Dies ermöglicht z.B. die Umsetzung von deutschen Umlauten in $\text{T}_{\text{E}}\text{X}$ - bzw. $\text{\LaTeX}$ -Kommandos.

Die Einstellungen für all diese Komponenten (z.B. Ausgabemodi und Formate, Grammatik, Präprozessoranweisungen und sonstige Programmeinstellungen) sind in einer Konfigurationsdatei gespeichert.

3.1 Die LEXPARSE-Strategie zur Segmentierung von Wörterbuchartikeln

Ein zentrales Problem beim Parsen von Wörterbüchern besteht in der korrekten Segmentierung der Wörterbuchartikel, d.h. im Erkennen der lexikographischen Angaben und deren Grenzen. Die LEXPARSE-Strategie unterscheidet sich deutlich von Ansätzen anderer Wörterbuchparser (wie z.B. Neff/Boguraev 1990 und Bläsi/Koch 1991).

Grundsätzlich werden beim LEXPARSE-System alle Wörter, Satzzeichen und Steuerzeichen des Satzbandes in einzelne Tokens zerlegt, d.h. daß eine aus mehreren Wörtern bestehende lexikographische Angabe zunächst mehrere Tokens umfasst.⁵ Dabei unterscheidet das System zwischen zwei verschiedenen Klassen von Tokens, den sog. *XCode-Tokens* und den sog. *Literal-Tokens*. Alle Strukturanzeiger (typographische und nicht-typographische) resultieren in XCode-Tokens, wohingegen alle anderen Satzzeichen, Wörter und Zahlen in Literal-Tokens resultieren.

Die Tokens werden vom Scanner an den Parser weitergegeben, der anhand der Grammatik überprüft, welche der Ableitungsregeln zu expandieren sind.⁶ Innerhalb der Ableitungsregeln lassen sich verschiedene Operatoren verwenden, die jeweils einzelne Tokens oder Folgen von Tokens lizensieren. Die durch die Operatoren lizensierten Tokens werden für die in der aktuellen Grammatikregel abzuleitenden Kategorie (die jeweils einer Klasse lexikographischer Angaben entspricht) in einem sog. *Akkumulator* angesammelt (akkumuliert). Kann die Kategorie vollständig abgeleitet werden, wird ihr der im Akkumulator angesammelte Angabentext zugewiesen.

5 Die Segmentierung in Tokens verläuft weitgehend nach herkömmlichen Regeln: d.h. Wortbegrenzungszeichen, Zeilenwechsel, Wechsel von Buchstabe zu Ziffer etc. resultieren jeweils in getrennten Tokens.

6 Im folgenden werden lediglich die grundlegenden Strategien des LEXPARSE-Systems vorgestellt. Für eine ausführliche Beschreibung von Syntax und Semantik des von LEXPARSE unterstützten Grammatikformalismus verweisen wir auf Hauser 1993 bzw. Hauser/Storrer 1993.

Durch die Aufteilung von Tokens in Tokenklassen und die Verwendung von verschiedenen Operatoren für diese unterschiedlichen Klassen lassen sich Ableitungsregeln für die verschiedensten lexikographischen Angaben spezifizieren.

Die drei zentralen Operatoren sind:

\$-Operator: Lizenziert genau ein Literal-Token.

Dieser Operator ist zur Beschreibung von lexikographischen Angaben geeignet, die aus nur einem Wort bestehen, wie z.B. die *Lemmzeichengestaltangabe* (LZGA), die durch Regel (1) erfasst werden kann:⁷

(1) LZGA -> \$.

\$\$-Operator: Lizenziert eine beliebige Folge von Literal-Tokens.

Dieser Operator ist zur Beschreibung von lexikographischen Angaben geeignet, die aus mehreren Wörtern (d.h. aus mehreren Literal-Tokens) bestehen und denen immer unmittelbar ein Strukturanzeiger folgt. Beim Duden-Stilwörterbuch sind einzelne *Beispielangaben* immer durch den Strukturanzeiger ‘;’ (Semikolon) voneinander getrennt; dies kann durch Ableitungsregel (2) erfasst werden:

(2) BeiA -> \$\$.

Der \$\$-Operator in (2) lizenziert eine beliebigen Folge von Literal-Tokens. Diese werden so lange in dem Akkumulator der Kategorie BeiA angesammelt, bis ein beliebiger XCode-Token in der Eingabe vorgefunden wird. Da das Semikolon im Duden-Stilwörterbuch die Funktion eines Strukturanzeigers besitzt, wird es vom Scanner immer als XCode-Token **XPSEM** (XCodePunctuationSEMicolon) generiert und beendet in diesem Fall die Expansion der o.a. Regel.

Da die Expansion aber auch durch jedes andere XCode-Token beendet wird, ist sichergestellt, daß die lexikographische Angabe **BeiA** durch alle in der Eingabe auftretenden Strukturanzeiger von anderen Angaben abgegrenzt wird, d.h. daß eine korrekte Segmentierung stattfindet.

***-Operator:** Lizenziert eine beliebige Folge von Tokens, d.h. sowohl Literal-Tokens als auch XCode-Tokens.

Dieser Operator ist zur Beschreibung von lexikographischen Angaben geeignet, die durch Strukturanzeiger begrenzt sind. So kann z.B. die *Pragmatische Angabe* durch die Regel (3) beschrieben werden:

(3) PragA -> XBRPO * XBRPC.

Die *Pragmatische Angabe* (**PragA**) wird durch runde Klammern begrenzt. Da diese Klammern als nicht-typographische Strukturanzeiger fungieren, werden sie vom Scanner als XCodes vom Typ **XBRPO** (XCodeBRacketParenthesisOpen) und **XBRPC** (XCodeBRacketParenthesisClose) generiert. Die Verwendung des *-Operator in

⁷ Alle hier exemplarisch beschriebenen lexikographischen Angaben finden sich auch in dem vom LEXPARSE-Programm generierten Ableitungsbaum im Anhang auf Seite 65, der einen Wörterbuchartikel des Duden-Stilwörterbuch beschreibt.

(3) erlaubt es, zwischen dem Auftreten des XCodes **XBRPO** und dem Auftreten des XCodes **XBRPC** eine beliebigen Folge von Tokens zu akkumulieren, die dann nach erfolgreicher Ableitung der Regel der Kategorie **PragA** zugewiesen werden. Durch die Verwendung des *****-Operators dürfen in dieser Folge von Tokens auch andere XCode-Tokens enthalten sein, d.h. andere Strukturanzeiger (wie z.B. ein Semikolon), die in diesem Kontext die Angabe noch weiter segmentieren, werden ignoriert.

3.2 Die Auflösung funktional ambiger Strukturanzeiger

Ein weiteres Kernproblem beim Parsen von Wörterbüchern besteht darin, daß ein und dasselbe Symbol — je nach Kontext — sowohl in strukturanzeigender Funktion als auch als Teil einer Angabe auftreten kann. Dieses Problem der funktionalen Ambiguität von Strukturanzeigern kann an der Verwendung des Kommas und des Doppelpunktes im Duden-Stilwörterbuch veranschaulicht werden:

- Das Komma trennt bei Nomina das Lemma von der Genusangabe und fungiert damit als nicht-typographischer Strukturanzeiger (*Treffen, das*). Ansonsten tritt es als normales Satzzeichen in Beispielen und Bedeutungserläuterungen auf (*regelmäßige, seltene Treffen*).⁸
- Der Doppelpunkt trennt beim Duden-Stilwörterbuch den *Formkommentar* vom *Semantischen Kommentar* und terminiert verschiedene andere Klassen lexikographischer Angaben.⁹ Der Doppelpunkt fungiert also typischerweise als nicht-typographischen Strukturanzeiger. Bei der genaueren Analyse des Wörterbuchs treten aber die Ausnahmen von dieser Regel zutage: In der *Beispielangabe* für das Lemma *trennen*: *die beiden Mannschaften trennten sich 0:0*, separiert der Doppelpunkt nicht zwei Angaben voneinander, sondern ist selbst Teil des Angabetextes.

Bei der Gestaltung einer LEXPARSE-Grammatik stellt sich nun die Frage, wie mit solchen Ambiguitäten zu verfahren ist, d.h. ob die Symbole ‘Komma’ und ‘Doppelpunkt’ auf entsprechende XCodes abgebildet werden sollen oder nicht. Die Beantwortung dieser Frage richtet sich danach, ob das betreffende Symbol überwiegend als Strukturanzeiger oder überwiegend als Teil von lexikographischen Angaben fungiert. LEXPARSE unterstützt die folgenden Strategien zur Lösung des Problems:

- Symbole, die überwiegend Teil von lexikographischen Angaben sind, werden nicht auf einen XCode abgebildet, sondern stattdessen direkt als Terminale in Ableitungsregeln integriert:

(4) LZGA -> \$ [", "^].

Mit Regel (4) kann die Kategorie *Lemmazeichengestaltangabe* (LZGA) zu genau einem Literal Token expandieren, dem fakultativ ein Komma folgen kann. Das Komma ist fakultativ (was durch die eckigen Klammern ausgedrückt wird), weil es in diesem Kontext ausschließlich bei Nomina auftritt. Das Häkchen (^) weist

⁸ Wobei die strukturanzeigende Funktion des Kommas innerhalb solcher elliptischen Konstruktionen bei unserer Betrachtung ignoriert wird.

⁹ Vgl. auch hierzu den Ableitungsbaum auf Seite 65.

das LEXPARSE-Programm an, das Komma in diesem Falle als Strukturanzeiger zu betrachten und es im Ableitungsbaum nicht darzustellen.¹⁰

- Symbole, die überwiegend als Strukturanzeiger fungieren, werden auf einen XCode abgebildet. Diese Strategie empfiehlt sich für Fälle wie dem des Doppelpunktes im Duden-Stilwörterbuch. Der Scanner generiert dann für den Doppelpunkt das XCode-Token XPCOL (*XCodePunctuationCOLon*) und erlaubt damit die Verwendung entsprechender Operatoren:

(5) WA → FK, XPCOL, SK.

(6) PragA → XBRPO * XBRPC.

Regel (5) expandiert die Kategorie *Wörterbuchartikel* (WA) zur Kategorie *Formkommentar* (FK), dem XCode XPCOL und der Kategorie *Semantischer Kommentar* (SK). Mit Regel (6) wird für die Kategorie *Pragmatische Angabe* (PragA) eine Folge beliebiger Tokens lizenziert, die durch runde Klammern (genauer deren XCode-Entsprechungen) begrenzt sind. Dabei kann diese Folge von Tokens ohne weiteres andere Strukturanzeiger (z.B. einen Doppelpunkt) bzw. deren XCode-Entsprechungen enthalten; auch diese werden vom *-Operator lizenziert.

Im Skopus einer *Beispielangabe* hingegen können — wie im o.g. Beispiel gezeigt wurde — Doppelpunkte auch Teil eines Angabetextes sein. In diesen Fällen bietet der LEXPARSE-Formalismus die Möglichkeit an, für den Skopus der entsprechenden Angabenklasse eine XCode-Ausnahme festzulegen:

(7) BeiA → -XPCOL: \$\$.

Die XCode-Ausnahme -XPCOL in (7) führt dazu, daß im Skopus der *Beispielangabe* (BeiA) der Doppelpunkt nicht als XCode-Token XPCOL, sondern als 'normales' Literal-Token ':' generiert wird. Er kann deshalb im Skopus von Regel (7) vom \$\$-Operator lizenziert und damit in den Akkumulator von BeiA integriert werden. Beispielsätze wie *die beiden Mannschaften trennten sich 0:0* können damit korrekt segmentiert werden.

3.3 Die Verarbeitung typographischer Information

Typographische Information wird in der Satzbanddatei durch Steuerzeichen kodiert, die einen Wechsel in eine bestimmte Schriftart oder in einen bestimmten Schriftschnitt anzeigen. Da es sich bei dieser Kodierung um typographische Strukturanzeiger handelt, werden diese Steuerzeichen als XCodes vom Typ XFT** (*XCodeFormatTypeface*) generiert.

Im Gegensatz zu einem Wörterbuchtext mit expliziter Markierung (wie z.B. einer SGML-Auszeichnung) ist der Skopus einer solchen Schriftumschaltung nicht nur durch die in der Satzbanddatei vorliegenden Steuerzeichen determiniert. Vielmehr gibt es noch andere Ereignisse, die den Wechsel in eine neue (oder alte) Schrift indizieren, wie z.B.:

- die Umschaltung auf ein Sonderzeichen,

10 Der Ableitungsbaum stellt die konkrete Mikrostruktur des Wörterbuchartikels dar und enthält deswegen keine Strukturanzeiger.

- das Ende eines Abschnitts,
- das Ende eines Wörterbuchartikels, etc.

Außerdem verwenden einige Setzer für ihre Satzbänder Makros, die einen Schriftwechsel für eine bestimmte Anzahl von Wörtern denotieren.¹¹

Es kann gezeigt werden, daß die Verteilung dieser Steuerzeichen im Satzband deterministisch nicht mit den Mitteln einer kontextfreien Grammatik beschrieben werden kann.¹² Deswegen verfügt das LEXPARSE-System über eine spezielle Einrichtung, die es erlaubt, die Ableitungsregeln der Grammatik zu attributieren. Zu diesem Zweck verwaltet LEXPARSE einen internen Zustandsanzeiger, der durch die o.a. Ereignisse automatisch auf den aktuellen Wert gesetzt wird, so daß der aktuelle Schriftzustand stets abgerufen werden kann. Die durch Schriftartenzustände attribuierten Ableitungsregeln können nur dann expandiert werden, wenn die durch die Attributierung beschriebene Schrift und der Wert des internen Zustandsanzeigers zum Zeitpunkt der Expansion übereinstimmen. Beispiele für eine derartige Regelattributierung finden sich in (8):

```
(8) S  -> XFTbo: A
      |   XFTit: B
      |   XFTst: C.
```

Kategorie S kann nur dann in Kategorie A expandiert werden, wenn der Zustandsanzeiger im Zustand 'bo' ist (der XCode XFTbo denotiert die Schriftart 'bold' [fett]), Kategorie S kann nur dann in Kategorie B expandiert werden, wenn der Zustandsanzeiger im Zustand 'it' ist (der XCode XFTit denotiert die Schriftart 'italics' [kursiv]), und analog dazu kann Kategorie S nur dann in Kategorie C expandiert werden, wenn der Zustandsanzeiger im Zustand 'st' ist (der XCode XFTst denotiert die Schriftart 'standard').

Die Möglichkeit zur Attributierung von Regeln bringt eine wesentliche Vereinfachung beim Schreiben einer Artikelstrukturgrammatik mit sich. Die Zustandsbedingungen erweitern den kontextfreien Formalismus und erlauben — im Hinblick auf die Verteilung von typographischen Strukturanzeigern — eine kontextsensitive Beschreibung.¹³

Auch über die wechselnde typographische Ausgestaltung bestimmter Angabeklassen kann dadurch hinweggegangen werden: Für solche Angabeklassen wird keine Zustandsbedingung formliert, d.h. der Schriftartenzustand ist für die Expansion der entsprechenden Ableitungsregel irrelevant.

11 Ein Beispiel eines solchen parametrisierten Makros wird beim Duden von den Setzern verwendet, um die Sperrung auszudrücken. Die Steuerzeichen `öSWy,xü` beschreiben eine Sperrung um `y` Prozent des normalen Weißraums für die nächsten `x` Worte.

12 Vgl. dazu Hauser 1993, Hauser/Storrer 1993. Werden typographische Strukturanzeiger als Terminale der Grammatik betrachtet — eine Betrachtungsweise, die im DEP-Ansatz implementiert ist und die auch das LEXPARSE-System optional unterstützt — können viele dieser terminalen Symbole nur fakultativ spezifiziert werden. Dadurch aber ist eine deterministische Ableitung nicht möglich.

13 Außerdem lassen sich auf diese Weise Inkonsistenzen in bezug auf die Verteilung der Steuerzeichen in der Satzbanddatei behandeln. So kommt es relativ häufig vor, daß z.B. bei einer kursiv gesetzten Angabe, die geklammert ist, die Steuerzeichen für die Umschaltung auf 'kursiv' in der Satzbanddatei abwechselnd einmal vor der öffnenden Klammer bzw. ein andermal nach der öffnenden Klammer gesetzt sind.

3.4 Technische Details

LEXPARSE wurde vollständig in der Programmiersprache C++ entwickelt. Durch die Verwendung von Konzepten der objektorientierten Programmierung wie Kapselung, Vererbung und Polymorphismus ist das Programm sehr übersichtlich, wartungsfreundlich und — auch aufgrund seiner Modularität — einfach zu erweitern. Darüber hinaus ist der Parser schnell und imstande, bis zu 10 Einträge pro Sekunde zu parsen.

Als Übersetzer wurde der BORLAND C++ Compiler für OS/2 Version 1.0 verwendet. Die Programmentwicklung fand unter dem Betriebssystem OS/2 2.0 auf einem IBM kompatiblen PC mit einer Intel '486 CPU statt. Das Programm wurde von Anfang an für eine möglichst einfache Portierung auf andere Plattformen konzipiert.

Das Programm ist quellcodekompatibel zum BORLAND C++ Compiler für DOS Version 3.1, zum GNU C++ Compiler Version 2.4.1. und zum AT&T C++ Compiler Version 3.0. Es läuft z.Zt. unter den Betriebssystemen MS-DOS, OS/2 und UNIX (SUN OS 4.1).

4 Schluß

Die Besonderheiten des LEXPARSE-Programmes lassen sich folgendermassen zusammenfassen:

- Zahlreiche Hilfsmittel unterstützen eine zügige und effiziente Grammatikentwicklung (wie z.B. Protokollmechanismen, Ausgabe maximaler Ableitungsbäume auch für Einträge, die nicht vollständig geparst werden konnten).
- Der Grammatikformalismus ist einfach zu erlernen.
- Der Grammatikformalismus verfügt über spezifische Zusatzkonstrukte, die auf das Parsen von Wörterbüchern hin zugeschnitten sind, wie z.B.:
 - Verwaltung von Schriftarten.
 - Verwaltung und Kontrolle von Zählern, die in Wörterbuchartikeln die verschiedenen homonymen und polysemen Lesarten eines Lemmas durchzählen und identifizieren.
 - Expansion von abgekürzten Lemmazeichen.
 - Behandlung und Erkennung von Fehlern in der Artikelstruktur.
- Die Ausgabekomponente ist benutzerkonfigurierbar und unterstützt z.Zt. drei verschiedene Modi:

Pfad-Notation: dieses Format ist nützlich für die Auswertung des Wörterbuchs durch Konkordanzprogramme.

Hierarchische Attribut-Werte-Struktur: dieses Format ist sehr gut zu lesen und kann deswegen zur schnellen Sichtung der geparsten Ableitungsbäume benutzt werden. (Vgl. Abbildung 1 (S. 56).)

SGML-getaggt: dieses Format dient dem Datenaustausch mit anderen Programmen bzw. der Weiterverarbeitung der geparsten Einträge.

Das LEXPARSE-Programm wurde bisher für folgende Wörterbücher eingesetzt:

- Duden-Stilwörterbuch: ca. 1600 Wörterbucheinträge; 90% aller Einträge sind wohlgeformt, 8% sind fehlerhaft, 2% konnten durch Grammatik nicht beschrieben werden.¹⁴
- Deutsches Rechtswörterbuch (Akademie der Wissenschaften in Heidelberg): deutlich über 80% aller Einträge können geparkt werden.
- Frühneuhochdeutsches Wörterbuch (Germanistisches Seminar der Universität Heidelberg): deutlich über 80% aller Einträge können geparkt werden.¹⁵

Zur Zeit wird LEXPARSE im Rahmen des EU-Projektes COMPASS am Seminar für Sprachwissenschaft der Universität Tübingen eingesetzt, in dem ein kontext-sensitives Nachschlageprogramm "Comprehension Assistant" für das Lesen von fremdsprachigen Texten entwickelt wird. Um mit LEXPARSE das im Projekt benutzte zweisprachige "Collins German-English Unabridged" zu parsen, waren nur minimale Erweiterungen am System notwendig.

5 Anhang

5.1 'Treffen', Duden-Stilwörterbuch, Seite 703

Als Beispiel einer LEXPARSE-Sitzung wurde der Wörterbuchartikel für das Lemma *Treffen* geparkt. Abgekürzte Lemmazeichen wurden vom Programm automatisch expandiert.

```
[D:\FOR\SFS\LEXPARSE]lexparse /entry -1 /nowarn lexparse.ini wbs\9-entries.dat
```

```
LEXPARSE V1.208 ß-Code   [Jul  1 1994]   <DEBUG-SRC>   A Dictionary Parser
```

```
(C) 1992-1994 Seminar fuer Sprachwissenschaft at the University of Tuebingen
```

```
(C) 1992-1994 Ralf Hauser
```

```
Created by 'Affie' (k.f. c|o)
```

```
LEXPARSE: Reading configuration: lexparse.ini
```

```
LEXPARSE: Grammar checked and OK.
```

```
Inputfile.....: wbs/9-entries.dat
```

```
Parsing.....: Enabled
```

```
Grammar nodes.....: 514
```

```
First entry.....: 1
```

```
Last entry.....: 1
```

```
Time of day.....: Fri Jul  1 10:54:07 1994
```

¹⁴ Eine entsprechende Grammatik für das Duden-Stilwörterbuch wurde von Sabine Engelke erstellt. Siehe dazu Engelke 1994.

¹⁵ Sowohl beim Deutschen Rechtswörterbuch, als auch beim Frühneuhochdeutschen Wörterbuch handelt es sich um handgesetzte Bücher, die aufgrund ihrer Entstehungsgeschichte Konsistenzprobleme verursachen.

```

-----
WA (Wörterbuchartikel)
+--> FK (Formkommentar)
| +--> LZGA (Lemmazeichengestaltangabe): "Treffen"
| +--> GrA (Grammatische Angabe)
|   +--> GA (Genusangabe): "das"
+--> SK (Semantischer Kommentar)
| +--> SSK1 (Semantischer Subkommentar 1. Stufe)
| | +--> PAA (Polysemieangabe (arabisch)): "1."
| | +--> PragSemA (Pragmatisch-Semantische Angabe)
| | | +--> BA (Bedeutungsangabe)
| | |   +--> BPA (Bedeutungsparaphrasenangabe): "Zusammenkunft, Begegnung"
| | +--> BeiGA (Beispielgruppenangabe)
| |   +--> BeiA (Beispielangabe): "regelmäßige, seltene Treffen"
| |   +--> BeiA (Beispielangabe): "ein Treffen der Abiturienten"
| |   +--> BeiA (Beispielangabe): "ein Treffen der Außenminister"
| |   +--> BeiA (Beispielangabe): "ein Treffen verabreden, veranstalten"
| |   +--> BeiA (Beispielangabe): "an einem Treffen teilnehmen"
| |   +--> BeiA (Beispielangabe): "zu einem Treffen kommen."
| +--> SSK1 (Semantischer Subkommentar 1. Stufe)
| | +--> PAA (Polysemieangabe (arabisch)): "2."
| | +--> PragSemA (Pragmatisch-Semantische Angabe)
| | | +--> PragA (Pragmatische Angabe): "militär. veraltet"
| | | +--> BA (Bedeutungsangabe)
| | |   +--> BPA (Bedeutungsparaphrasenangabe): "Gefecht"
| | +--> BeiGA (Beispielgruppenangabe)
| |   +--> BeiA (Beispielangabe): "frische Truppen ins Treffen führen."
| +--> SSK1 (Semantischer Subkommentar 1. Stufe)
| | +--> PAA (Polysemieangabe (arabisch)): "3."
| | +--> PragSemA (Pragmatisch-Semantische Angabe)
| | | +--> PragA (Pragmatische Angabe): "Sport"
| | | +--> BA (Bedeutungsangabe)
| | |   +--> BPA (Bedeutungsparaphrasenangabe): "Wettkampf"
| | +--> BeiGA (Beispielgruppenangabe)
| |   +--> BeiA (Beispielangabe): "ein faires, spannendes Treffen"
| |   +--> BeiA (Beispielangabe): "das Treffen endete unentschieden"
| |   +--> BeiA (Beispielangabe): "sie konnte das Treffen für sich entscheiden."
+--> PKP (Postkommentar zur Phraseologie)
| +--> SKP (Subkommentar zur Phraseologie)
| | +--> PragA (Pragmatische Angabe): "geh."
| | +--> PhrasA (Phrasenangabe): "etwas ins Treffen führen"
| | +--> KPB (Kommentar zur Phrasenbedeutung)
| |   +--> PBA (Phrasenbedeutungsangabe): "etwas als Argument vorbringen"

```

Entry #1: Accepted!

```

-----
Inputfile.....: wbs/9-entries.dat
First entry.....: 1
Last entry.....: 1
Entries parsed.....: 1
Errors.....: 0
Time of day.....: Fri Jul 1 10:54:07 1994
Time of start.....: Fri Jul 1 10:54:07 1994
Elapsed time.....: 00h 00m 00s
Average parse time.: < 1 sec.
LEXPARSE V1.208 ß-Code [Jul 1 1994] <DEBUG-SRC>
Program terminated successfully

```

Literatur

- Brigitte Bläser & Matthias Wermke (1990): *Projekt 'Elektronische Wörterbücher / Lexika' Abschlußbericht der Definitionsphase*. IWBS Report 145. Wissenschaftliches Zentrum / Institut für Wissensbasierte Systeme der IBM Deutschland GmbH.
- Christoph Bläsi & Heinz-Detlev Koch (1991): *Maschinelle Strukturerschließung von Wörterbuchartikeln mit Standardmethoden*. In: *Lexicographica*. Vol. 7, (1991) 1992, S. 182–201. Tübingen: Niemeyer.
- Bran Boguraev (1991): *Building a Lexicon: The Contribution of Computers*. In: *International Journal of Lexicography* Vol. 4, (1991), S. 227–260
- & Ted Briscoe (Hrsg.) (1989): *Computational Lexicography for Natural Language Processing*. London, New York: Longman.
- Ted Briscoe (1991): *Lexical Issues in Natural Language Processing*. In: *Natural Language and Speech*. Hrsg.: Klein, E. & Veltmann, F. Symposium Proceedings, Berlin, Heidelberg, New York, 39–69.
- Duden-2: *Das Stilwörterbuch* (1988). G. Drosdowski (Hrsg.) 7. neu bearb. und erw. Auflage, Bibliographisches Institut Mannheim.
- Duden-10: *Das Bedeutungswörterbuch* (1985). W. Müller (Hrsg.) 2. neu bearb. und erw. Auflage, Bibliographisches Institut Mannheim.
- Sabine Engelke (1994, in Vorbereitung): *Grammatikentwicklung für Wörterbücher mit dem LEXPARSE Formalismus*. Sfs Report, Seminar für Sprachwissenschaft der Universität Tübingen.
- Ralf Hauser (1993): *LEXPARSE User's Manual*. Sfs Report 10–93, Seminar für Sprachwissenschaft der Universität Tübingen.
- & Angelika Storrer (1993): *LEXPARSE — Ein Parser zur maschinellen Analyse von Wörterbuchstrukturen*. Sfs Report 9–93, Seminar für Sprachwissenschaft der Universität Tübingen.
- & Angelika Storrer (1994): *Dictionary Entry Parsing Using the LEXPARSE System* In: *Lexicographica*. Vol. 9, (1993) 1994, S. 174–219. Tübingen: Niemeyer.
- Franz Josef Hausmann, Oskar Reichmann, Herbert Ernst Wiegand & Ladislav Zgusta (Hrsg.) (1989): *Wörterbücher. Ein internationales Handbuch zur Lexikographie*. Berlin: de Gruyter.
- & Herbert Ernst Wiegand (1989): *Component Parts and Structures of General Monolingual Dictionaries: A Survey*. In: *Wörterbücher. Ein internationales Handbuch zur Lexikographie*. Hrsg.: Franz J. Hausmann, Oskar Reichmann, Herbert E. Wiegand & Ladislav Zgusta. Berlin: de Gruyter, Vol. 1, S. 328–360.
- John McNaught (1988): *Computational Lexicography and Computational Linguistics* In: *Lexicographica*. Vol. 4, 1988, S. 19–33. Tübingen: Niemeyer.
- Mary S. Neff & Branimir K. Boguraev (1990): *From Machine-Readable Dictionaries to Lexical Data Bases*. Research Report RC #16080 (71353) 8/31/90, IBM Thomas J. Watson Research Center, Yorktown Heights, New York.
- Angelika Storrer, Helmut Feldweg & Erhard Hinrichs (1993): *Korpusunterstützte Entwicklung lexikalischer Wissensbasen*. In: *Sprache und Datenverarbeitung* Vol. 17, (1993) Bonn 1994, S. 59–72.
- Herbert Ernst Wiegand (1989b): *Der Begriff der Mikrostruktur: Geschichte, Probleme, Perspektiven*. In: *Wörterbücher. Ein internationales Handbuch zur Lexikographie*. Hrsg.: Franz J. Hausmann, Oskar Reichmann, Herbert E. Wiegand & Ladislav Zgusta. Berlin: de Gruyter, Vol. 1, S. 409–462.
- (1989c): *Formen von Mikrostrukturen im allgemeinen einsprachigen Wörterbuch*. In: *Wörterbücher. Ein internationales Handbuch zur Lexikographie*. Hrsg.: Franz J. Hausmann, Oskar Reichmann, Herbert E. Wiegand & Ladislav Zgusta. Berlin: de Gruyter, Vol. 1, S. 462–501.
- (1991): *Printed Dictionaries and their Parts as Text. An Overview of More Recent Research as an Introduction* In: *Lexicographica*. Vol. 6, (1990) 1991, S. 1–124. Tübingen: Niemeyer.